PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In the **PATENT APPLICATION** of: Brooks et al.	
Application No.: 09/774,545	Our File: TAN-2-1472.01.US
Confirmation No.: 3228	Date: **April 30, 2009**
Filed: January 31, 2001	
For: ADAPTIVE COMPRESSION IN AN EDGE ROUTER	
Group: 2451	
Examiner: Hassan A. Phillips	

APPEAL BRIEF

Mail Stop Appeal Briefs-Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

This Brief is submitted in response to the Notice of Panel Decision from Pre-Appeal Brief mailed on April 21, 2009.

TABLE OF CONTENTS

I. Real Party in Interest

The real part in interest is the assignee of record IPR Licensing Inc., which is controlled by InterDigital Communications Corporation a/k/a InterDigital, having corporate headquarters at 781 Third Avenue, King of Prussia, PA 19406.

II. Related Appeals and Interferences

There are no appeals or interferences related to this patent application.

III. Status of Claims

Claims 1-7, 9-19, 21-24, 28 and 32 are pending in this application. Claims 1-7, 9-19, 21-24, 28 and 32 are all appealed and currently stand rejected based on the November 4, 2008 Final Office Action. Claims 1-7, 9-19, 21-24, 28 and 32 are the subject of this Appeal and are reproduced in the attached Claims Appendix.

IV. Status of Amendments

The last claim amendment was made in a Reply dated August 19, 2008.

V. Summary of Claimed Subject Matter

The present application contains three independent claims 1, 13, and 28. Dependent claims 2-7, 9-12 and 32 depend from independent claim 1; dependent

claims 14-19 and 21-24 depend from independent claim 13. A summary of each independent claim citing support from the application's U.S. Publication No. 2002/0103938 follows.

Independent claim 1 is directed to a method for optimizing compression efficiency in data packet communications. Specification ¶¶ [0007]-[0008], ¶¶ [0087-0094], ¶¶ [0099-0105], Figures 9A, 9B, 10 and 12. Claim 1 defines an indirect process for filtering a protocol data unit (PDU) to determine compressibility of its data contents. Specification ¶¶ [0008] and specific example at ¶[0089], Figures 9A and 9B. The process of Claim 1 is indirect since it requires "tracking previously filtered [PDUs] and information regarding the compression applied to previously filtered [PDUs]." See specific example at ¶[0090], Figure 9B "Stream Association Table." Where the PDU is associated with a previously filtered PDU, the data link compression that was applied to the previously filtered PDU is selected and used "to control a compression process" for the PDU. This is an indirect way of determining the PDU's data type, since it relies on the prior determination made of the contents of the previously filtered PDU. See specific example at ¶¶[0091]-[0092], Figures 9A and 9B where compression is disabled for PDU 5 that is associated with stream 2 that starts with PDU 3 that was previously

determined to have JPEG data and compression is enabled for PDUs 2 and 4 that are associated with stream 1 that starts with PDU 1 that was previously determined to have Text data.

Independent claim 13 is directed to an apparatus configured to implement the process of claim 1. Specification ¶¶ [0007]-[0008], ¶¶ [0068-0074], ¶¶ [0087-0094], ¶¶ [0099-0105], Figures 6A, 6B, 9A, 9B, 10 and 12. Claim 13 defines a PDU filter, e.g. 610/625 Figures 6A/B, a tracking unit, e.g. Figure 9B "Stream Association Table" 905, and a selector, e.g. 615 Figures 6B. The filter indirectly filters a PDU to determine compressibility of its contents by determining if it is associated with a previously filtered PDU. The tracker enables the filter to perform such indirect filtering, by tracking the compression applied to previously filtered PDUs. The selector selects the compression that was applied to a previously filtered PDU where the PDU is associated with the previously filtered PDU. The selector uses the compression selection to control a compressor 615, for example by enabling or disabling compression of the PDU. See ¶¶[0091]-[0092].

Independent claim 28 is directed to a computer-readable medium having stored instructions to implement a method similar to claim 1, but more limited. Specification ¶¶ [0007]-[0008], ¶¶ [0087-0094], ¶¶ [0099-0105], Figures 6A, 6B,

9A, 9B, 10 and 12. The implemented method includes the indirect filtering of a PDU to determine compressibility of its contents by determining if it is associated with a previously filtered PDU. In order to perform such filtering, claim 28 requires tracking the compression applied to previously filtered PDUs. Claim 28, however, also, defines direct filtering by "determining the type of data of the [PDU] where the given protocol data unit is not associated with a previously filtered protocol data unit." Thus, if it is first determined that indirect filtering can not be performed, because there is no associated previously filtered PDU, a direct determination of data type is made. See ¶[0090]; also see step 1245 of Figure 12. Regarding the example of Figures 9A and 9B, the compression for stream 1 is determined as Text based on directly determining the data type of PDU 1 that starts stream 1; the compression for stream 2 is determined as JPEG based on directly determining the data type of PDU 3 that starts stream 2. Where the PDU is associated with a previously filtered PDU indirect filtering is used so that the data link compression that was applied to the previously filtered PDU is selected and used to the control compression process for the PDU. See ¶¶[0091]-[0092]; also see ¶¶[0103]-[0105], Figure 12.

VI. Grounds of Rejections to be Reviewed on Appeal

Claims 1, 2, 6, 7, 9-14, 18-19, 21-24, 28 and 32 are rejected as anticipated by U.S. Patent No. 5,838,927 (Gillon). This rejection cites Gillon's teachings at Column 5, line 33 to Column 6, line 6; Column 7, lines 6-9; and Figures 4A and 6.

It is unclear whether the Examiner bases anticipation of the claimed indirect "filtering" by "tracking previously filtered [PDUs] and information regarding the compression applied to previously filtered [PDUs]," based on the LZP compression algorithm referenced in Column 5, lines 33-38 or the general description of Figure 4a referenced in Column 5, lines 48-57. For example page 3 of the Final Action states:

> ... Gillon suggests a filter accessing a table having entries with specific media types deemed compression limited to associate individual PDU's to a specific media type, where Gillon teaches examining a packet prior to compression and using the header of the packet to determine whether data following the header can be compressed based on the type of data following the header, (see Gillon, col. 5, lines 48-57). ... Contrary to applicant's arguments however, examiner also maintains the teachings of Gillon suggest the teachings in the second sentence of par. [0008], and thus claim I, where Gillon teaches using compression algorithms such as LZP, (Gillon, col. 5, lines 33-38). ...

With respect to claims 9 and 21 that are directed to using a table, e.g. Figure 9B "Stream Association Table" 905, to perform the claimed indirect filtering, the

Examiner asserts that such a table is "inherent" citing the general description of Figure 4a at page 6 of the Final Action stating:

> In considering claims 9 and 21, it is inherent in the method taught by Gillon that a table is accessed including tracking information about previously filtered PDU's, when determining if a given PDU is associated with a previously filtered PDU. See col. 5, lines 48-56.

With respect to dependent claims 18 and 32, that are similar to more limited independent claim 24, the Examiner contends the alternative treatment of PDUs in determining data type is "inherent," stating at page 7 of the Final Action:

> In considering claims 18 and 32, it is inherent in the apparatus and method taught by Gillon that the filter is configured to determine compressibility of the contents of the given protocol data unit by determining the type of data of the given protocol data unit where the given protocol data unit is not associated with a previously filtered protocol data unit, (col. 5, lines 33-38, col. 5, lines 48-57, and col. 7, lines 6-9); and the selector is configured to select the state of data link compression for the given protocol data unit based on the determined type of data of the given protocol data unit if the given protocol data unit is not associated with a previously filtered protocol data unit, (col. 5, lines 33-38, col. 5, lines 48-57, and col. 7, lines 6-9); wherein the selector includes a table configured to store entries with specific media types deemed compression limited, (col. 5, lines 39-50).

Review of the rejection of dependent claims 3-5 and 15-17 as obvious over Gillon in view of U.S. Patent No. 5,555,377 (Christensen) is not in itself sought. However, withdrawal of this rejection is sought based on the allowability of the independent claims as discussed below.

VII. Argument

Claims 1, 2, 6, 7, 9-14, 18-19, 21-24, 28 and 32 stand rejected under 35 U.S.C. §102 as anticipated by Gillon. Applicants respectfully disagree.

A. The Teachings Of Gillon

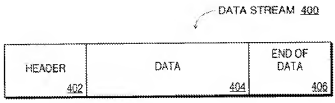
Gillon teaches improvements in compression processing of Data Units that include data portions of indeterminate length, called Data Streams or Data Packets. This is contrasted to prior art compression processing of Data Units that include data portions of a predetermined length, called Data Blocks.

Gillon uses the term "Data Stream" synonymously with "Data Packet." For example, Column 5, lines 39-57 states:

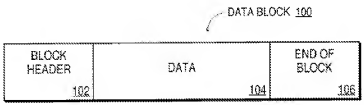
> As illustrated in FIG. 4A, **data stream 400** includes a header 402, data 404 and end of data indicator 406. ...
>
> **Data packet 400** is examined prior to compression, and header 402 is used to determine whether data 404 can be compressed. According to one embodiment of the present invention, data 404 is HTML data, as indicated by content type header "text/HTML." When the compression unit detects **data packet 400** with a content header that indicates that data 404 is compressible, data 404 is immediately attached to a compression stream that transmits the compressed data. Note that other predetermined functions may be performed on **data packet 400** prior to compression. ... (Emphasis added)

No where in Gillon is the term "Data Stream" used to indicate any type of series of "Data Packets" or other "Data Units" that are associated with each other.

Gillon Figure 4a depicts a "DATA STREAM 400" as:



This is similar to the depiction of the prior art "DATA BLOCK 100" illustrated in Gillon Figure 1A as:

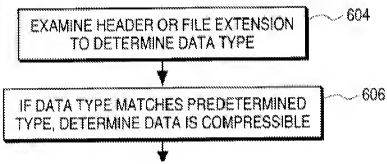


The difference between a Data Block 100 and a Data Steam a/k/a/Data Packet 400, as defined by Gillon, is that the Data portion 104 of the Data Block 100 is of a predetermined length and the Data portion 404 of the Data Steam a/k/a Data Packet 400 as of indeterminate length. Generically, a Data Block 100 or a Data Steam a/k/a Data Packet 400, as defined by Gillon, can be considered a "Data Unit," the term used in the appealed claims.

Gillon is directed to compression processing of Data Units having indeterminate-length data portions, i.e. Data Streams/Packets using Gillon's terminology. The type of compression processing taught by Gillon is "continuous compression processing" as illustrated in Gillon Figures 4B and 4C, as opposed to

the prior art "discontinuous compression/transmit processing" of "Data Blocks" illustrated in Gillon Figure 1B. Instead of performing compression and transmission at the end of each Data Block 100, Gillon teaches the creation of a "compression stream" to permit continuous processing. The "compression stream" is a series of Gillon's Data Streams/Packets for which a determination has been made that their data is compressible. This "continuous compression processing," according to Gillon, avoids latency problems associated with the prior art "discontinuous compression/transmit processing."

Gillon Figure 6 depicts general processing steps for data streams/packets relevant to the appealed claims, in particular, steps 604 and 608 reproduced below:



The explanatory text of Figure 6 appears at Column 7, lines 3-15 that states:

> FIG. 6 is a flowchart that illustrates one embodiment of the present invention. Based on a client request, a data stream is received from a remote source or retrieved from a local disk in step 602. **In step 604, the header of the data or the file extension is examined to determine the data type**. If the data type matches a predetermined type, in step 606 the data is determined to be compressible. The data

> stream is attached to a compression stream or any other type of steam such as an encryption stream in step 608. The compression stream and other streams are then attached to a write stream in step 610. In step 612, write stream is transmitted with the attached streams, and in step 614 the write stream is received by the client. (Emphasis added)

The determining step 604 is performed directly on the Data Stream/Packet in one of two ways: examining the header or examining the file type. These two alternatives are explained in detail in Gillon at Column 5, line 48 - Column 6, line 4 that state:

> Data packet 400 is examined prior to compression, and header 402 is used to determine whether data 404 can be compressed. **According to one embodiment of the present invention, data 404 is HTML data, as indicated by content type header "text/HTML." When the compression unit detects data packet 400 with a content header that indicates that data 404 is compressible, data 404 is immediately attached to a compression stream that transmits the compressed data.** Note that other predetermined functions may be performed on data packet 400 prior to compression. These predetermined functions include counting the bytes in the data stream and printing the bytes in the data stream.
>
> **According to another embodiment of the present invention, data stream 400 does not include a header 402. In this embodiment, the file extension of the retrieved data may be examined to determine what type of data may be found in that file.** For example, even if the file is an HTML file without a content type header "text/HTML," the type of data in the file may still be determined by examining the file extension. In this example, if the file extension is .HTML, the type of data in the file is determined to be HTML data, and the data in the file is marked as compressible. Other file extensions include AVI for a Microsoft digital video file, for example. ... (Emphasis added)

The determination step 604 in both cases seeks to directly determine data type for

the Data Steam/Packet 400 based on information in the Data Steam/Packet 400 that directly identifies the type of data. In the first case, an identifying "content header" is examined; in the second case, an identifying "file extension" is examined. Gillon is silent as to how a determination can be made without some direct reference indicating the data type in the Data Stream/Packet itself.

B. The Standard for Anticipation

It is well established that the Examiner has the initial burden of establishing a *prima facie* case of anticipation by establishing that a single prior art reference discloses, expressly or under the principles of inherency, each and every element of a claimed invention as well as disclosing structure which is capable of performing the recited functional limitations. Ex parte Gross and Spitaleri, 2002 Pat. App. LEXIS 218, *3 (B.P.A.I. 2002) (citing In re Oetiker, 977 F.2d 1443, 1445, 24 USPQ2d 1443, 1444 (Fed. Cir. 1992)); RCA Corp. v. Applied Digital Data Systems, Inc., 730 F.2d 1440, 1444, 221 USPQ 385, 388 (Fed. Cir.1984). Under the doctrine of inherency, extrinsic evidence "must make clear that the missing descriptive matter is necessarily present in the thing described in the reference, and that it would be so recognized by persons of ordinary skill." Continental Can Co. v. Monsanto Co., 948 F.2d 1264, 1268, 20 U.S.P.Q.2d 1746,

1749 (Fed. Cir. 1991). "Inherency, however, may not be established by probabilities or possibilities. The mere fact that a certain thing may result from a given set of circumstances is not sufficient." Continental Can at 1269 (*quoting* In re Oelrich, 666 F.2d 578, 581, 212 U.S.P.Q. 323, 326 (C.C.P.A. 1981)); Ex parte Richard Brothers, 2002 Pat. App. LEXIS 297 (B.P.A.I. 2002); Ex parte Morris, 2001 Pat. App. LEXIS 111 (B.P.A.I. 2001).

Applicants respectfully submit that the Final OA fails to establish a *prima facie* case of anticipation since Gillon does not disclose or suggest the claimed indirect filtering or any structure capable of performing the claimed indirect filtering, such as the "Stream Association Table" depicted in applicants' Figure 9B.

C. Gillon Does Not Anticipate Any of the Claims

The pending claims are directed to "filtering" Data Units[1] identified as Protocol Data Units (PDUs). Such "filtering" is done to determine the type of data in the data portion of a Data Unit. This "filtering" used to determine whether or not to compress the data for transmission, since some types of data are

[1] This language is generic to Data Units having either an indeterminate data length or a predetermined data length, i.e. to both Data Blocks and the Data Streams/Packets as defined by Gillon.

compressible and other types of data are not. This is generally the objective of Step 604 of Gillon Figure 6.

1. Independent Claims 1 and 7 - Indirect Filtering

Unlike the direct data-type determination taught by Gillon, independent claims 1 and 7 define "filtering" for a PDU based upon an association with a previously filtered PDU, i.e. indirect filtering. This introduces an initial step of determining the existence of such an association. Claim 1, for example, recites:

> ... filtering protocol-specific header and control information of a protocol data unit (PDU) to determine compressibility of the contents of said protocol data unit **including determining if a given protocol data unit is associated with a previously filtered protocol data unit by tracking previously filtered protocol data units and information regarding the compression applied to previously filtered protocol data units**;
>
> based on the result of said filtering, selecting the state of data link compression for said protocol data unit to optimize compression efficiency such that **if the given protocol data unit is associated with a previously filtered protocol data unit, the data link compression that was applied to the previously filtered protocol data unit is selected**; ....

In Gillon, the "filtering," i.e. the determination of the data type, is done by obtaining data type information directly from the Data Unit either from a direct indication in the header or direct examination of the file extension contained in the Data Unit itself. Gillon Step 604, Column 5, line 48 - Column 6, line 4. Unlike

Gillon, the claims define indirect "filtering" through first determining if the Data Unit is associated with a previously filtered Data Unit and, if so, then using the prior determination made for the previously filtered Data Unit that was based on the information contained in the previously filtered Data Unit. Accordingly, the contents of the Data Unit are not directly relied on in making the data-type determination as taught by Gillon.

Gillon does not teach or suggest making any such indirect determination of the data type as defined by independent claims 1 and 7. There is no tracking of associations of Data Units in order to be able to use a prior data-type determination in place of the direct data-type determination taught by Step 604 of Gillon. Claims 1 and 7 define the re-use of prior determinations of data type of an associated Data Unit; a concept not even intimated in Gillon. Accordingly, the rejection based on anticipation by Gillon should be reversed.

2. Claims 18, 24 and 32 - Indirect Filtering Combined With Direct Filtering

Dependent claims 18 and 32 and independent claim 24 define the use of both indirect and direct "filtering." Indirect "filtering" is defined with respect to where there is the existence of an associated previously filtered PDU as discussed above. Direct "filtering" is used for the case where the PDU is not associated with a

previously filtered PDU. For example, claim 32 which depends on claim 1 defines the direct determination of data type stating:

> ... determining the type of data of the given [PDU] unit where the given [PDU] is not associated with a previously filtered protocol data unit;

This determining step of dependent claims 18 and 32 and independent claim 24, could be performed by the direct type of data determination taught by Gillon Step 604. However, there is no disclosure of suggestion of the precursor step of ascertaining whether an indirect or a direct data-type determination should be used as required by the claimed "determining if a given protocol data unit is associated with a previously filtered protocol data unit."

Where there is an association with a previously filtered PDU, the independent claims require the indirect determination of the data type of the PDU, not based directly on the PDU itself, but based on the prior determination made for the associated previously filtered PDU. Not only is the claimed indirect determination step and function entirely absent from Gillon, but also Gillon fails to disclose or suggest the claimed combination of selecting an indirect data-type determination for some Data Units and a direct data-type determination for others as defined by dependent claims 18 and 32 and independent claim 24.

3. Claims 9 and 21 - Indirect Filtering Using A Tracking Table

An example of the claimed indirect filtering by tracking previously filtered PDUs is provided in application Figures 9A and B reproduced below:

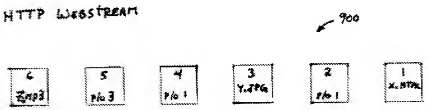


FIG. 9A

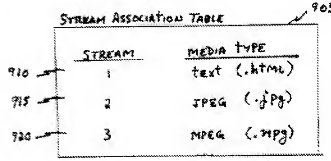


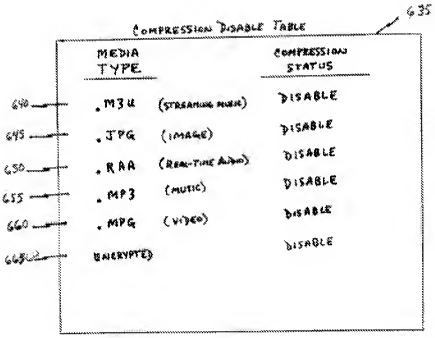
FIG. 9B

The "Stream Association Table" is an example of structure used to implement the claimed indirect data-type determination that is entirely absent from Gillon. Dependent claims 9 and 21 reference such a table for which there is no counterpart disclosure in Gillon.

Figure 9a illustrates six PDUs of a series in an HTTP webstream that include substreams of different data types. A first substream starts with PDU 1 that is determined to be text; a second substream starts with PDU 3 that is determined to

be JPEG. The claimed filtering process tracks associations with previously filtered packets, such as by creating the Stream Association Table, as defined by claims 9 and 21. When PDU 2 is filtered, it is first determined that PDU 2 is associated with PDU 1, then reference to the Stream Association Table is made to conclude that PDU 2's data type is text without any further examination of PDU 2. Similarly, when PDU 5 is filtered, it is first determined that PDU 5 is associated with PDU 3, then PDU 5 is assumed to be JPEG without any further examination of PDU 5, since PDU 3 was previously filtered and determined to be JPEG.

The "Stream Association Table" is not the same as a "compression table," such as the table illustrated in applicants' Figure 6c, reproduced below.



As explained at ¶¶ [0075]-[0076] in the published application, the "Compression Disable Table" is referenced after making the data-type determination for the Data Unit and is used to enable or disable the compression process based on the determined type of data. Such a compression table is not relevant to the initial filtering determination of what type of data is contained in the Data Unit. Such a compression table is not equivalent to or used to perform the same function as the "Stream Association Table" of applicants' Figure 9B.[2]

A compression table could be referenced to perform Step 606 of the Gillon Figure 6 method. However, such a table would not play a role in the initial data-type determination of Step 604 of Gillon Figure 6. Even if one would accept the contention that some type of a "Compression Table" is inherently taught by Step 606 of Gillon Figure 6, this does not disclose or suggest the indirect filtering or the function of the "Stream Association Table" taught and claimed by applicants.

The Examiner's assertion that a "Stream Association Table" is "inherent" is without merit. The absence of any counterpart of such structure in Gillon for implementing the claimed indirect filtering defeats the Examiner's anticipation

[2] This is not to imply that the two types of tables could not be combined.

rejection of claims 7 and 21 as well as the anticipation rejection of the independent claims. Ex parte Gross and Spitaleri, supra.

4. The Examiner's Remaining Contentions Are Without Merit

The Examiner's further argument that using LZP compression as taught by Gillon somehow anticipates the claims is also without merit. While it is true that LZP compression applies a compression dictionary that was built from previously compressed data packets, this has nothing to do with the filtering of packets to determine whether or not they should be processed by the LZP algorithm. The claimed tracking is part of the indirect filtering to determine if a PDU should be subject to compression. The LZP algorithm is directed to the compression processing after an appropriate filtering determination has been made.

Gillon simply does not teach any of the claimed subject matter associated with indirect filtering to determine a Data Unit's data type. Accordingly, Gillon does not anticipate any of the pending claims.

D. Conclusion

For the above reasons, reversal of the anticipation rejection of claims 1, 2, 6, 7, 9-14, 18-19, 21-24, 28 and 32, withdrawal of the rejection of dependent claims 3-5 and 15-17 as moot, and allowance are respectfully requested.

Respectfully submitted,

Brooks et al.

By__/C. Frederick Koenig III/____
C. Frederick Koenig III
Registration No. 29,662

Volpe and Koenig, P.C.
United Plaza, Suite 1600
30 South 17th Street
Philadelphia, PA 19103
Telephone: (215) 568-6400
Facsimile: (215) 568-6499

CFK/lhe

VIII. Claims Appendix

1. A method for optimizing compression efficiency comprising:

filtering protocol-specific header and control information of a protocol data unit (PDU) to determine compressibility of the contents of said protocol data unit including determining if a given protocol data unit is associated with a previously filtered protocol data unit by tracking previously filtered protocol data units and information regarding the compression applied to previously filtered protocol data units;

based on the result of said filtering, selecting the state of data link compression for said protocol data unit to optimize compression efficiency such that if the given protocol data unit is associated with a previously filtered protocol data unit, the data link compression that was applied to the previously filtered protocol data unit is selected; and

associating the selected state of data link compression with the protocol data unit to control a compression process adapted to compress contents of protocol data units.

2. The method as claimed in claim 1, further including compressing the

contents of the protocol data unit as a function of the state of data link compression.

3. The method as claimed in claim 2, wherein compressing the contents of the protocol data unit includes applying an indication in or with the compressed protocol data unit to indicate whether the contents of the protocol data unit have been compressed.

4. The method as claimed in claim 3, further including decompressing the compressed contents of the protocol data unit.

5. The method as claimed in claim 4, wherein, based on the indication of whether the contents of the protocol data unit have been compressed, decompressing the compressed contents of the protocol data unit is performed in a manner previously negotiated.

6. (Original) The method as claimed in claim 1, further including accessing a table having entries with specific media types deemed compression

limited.

7. The method as claimed in claim 1, wherein filtering includes associating individual protocol data units to a specific media type.

8. Cancelled.

9. The method as claimed in claim 1, wherein determining includes accessing a table including tracking information about previously filtered protocol data units.

10. The method as claimed in claim 1, wherein selecting the state of the data link compression includes disabling the data link compression if the compressibility of the contents of the protocol data unit is determined to be low.

11. The method as claimed in claim 1, wherein selecting the state of the data link compression includes enabling the data link compression if the compressibility of the contents of the protocol data unit is determined to be high.

12. The method as claimed in claim 1, further including initializing a table used by the data link compression with data patterns expected to be contained in the content of said protocol data unit.

13. An apparatus for optimizing compression efficiency comprising:

a protocol data unit (PDU) filter configured to determine compressibility of the contents of a given protocol data unit by determining if the given protocol data unit is associated with a previously filtered protocol data unit;

a tracking unit operatively associated with the PDU filter configured to track previously filtered protocol data units to enable the PDU filter to determine if the given protocol data unit is associated with a previously filtered protocol data unit; and

a selector coupled to the output of the filter and configured (i) to select the state of data link compression for the protocol data unit to optimize compression efficiency such that if the given protocol data unit is associated with a previously filtered protocol data unit, the data link compression that was applied to the previously filtered protocol data unit is selected; and (ii) to associate the selected state of data link compression with the protocol data unit to control a compressor

adapted to compress contents of protocol data units.

14. The apparatus as claimed in claim 13, further including a compressor configured to compress the contents of the protocol data unit responsive to the state of data link compression.

15. The apparatus as claimed in claim 14, wherein the compressor is configured to include an indication in or with the compressed protocol data unit to indicate whether the contents of the protocol data unit have been compressed.

16. The apparatus as claimed in claim 15, further including a decompressor configured to decompress the compressed contents of the protocol data unit.

17. The apparatus as claimed in claim 16, wherein, the decompressor is configured to decompress the contents of the protocol data unit in a manner previously negotiated with the compressor based on the indication of whether the contents of the protocol data unit have been compressed.

18. The apparatus according to claim 13, wherein:

the filter is configured to determine compressibility of the contents of the given protocol data unit by determining the type of data of the given protocol data unit where the given protocol data unit is not associated with a previously filtered protocol data unit;

the selector is configured to select the state of data link compression for the given protocol data unit based on the determined type of data of the given protocol data unit if the given protocol data unit is not associated with a previously filtered protocol data unit; and

the selector includes a table configured to store entries with specific media types deemed compression limited.

19. The apparatus as claimed in claim 13, wherein the filter is configured to associate individual protocol data units to a specific media type.

20. Cancelled.

21. The apparatus as claimed in claim 13, wherein the filter further

includes a table configured to store information about previously filtered protocol data units.

22. The apparatus as claimed in claim 13, wherein the selector is configured to disable the compressor if the compressibility of the contents of the protocol data unit is determined to be low.

23. The apparatus as claimed in claim 13, wherein the selector is configured to enable the compressor if the compressibility of the contents of the protocol data unit is determined to be high.

24. The apparatus as claimed in claim 13, further including an initialization unit configured to initialize a table used by the compressor with data patterns expected to be contained in the content of said protocol data unit.

25.-27. Cancelled.

28. A computer-readable medium having stored thereon sequences of

instructions, the sequences of instructions including instructions, when executed by a processor, configured to cause the processor to perform:

filtering protocol-specific header and control information of a protocol data unit to determine compressibility of the contents of said protocol data unit including:

determining if a given protocol data unit is associated with a previously filtered protocol data unit by tracking previously filtered protocol data units and information regarding the compression applied to previously filtered protocol data units; and

determining the type of data of the given protocol data unit where the given protocol data unit is not associated with a previously filtered protocol data unit;

selecting the state of data link compression for said protocol data unit based on the results of said filtering to optimize compression efficiency such that:

if the given protocol data unit is associated with a previously filtered protocol data unit, the data link compression that was applied to the previously filtered protocol data unit is selected; and

otherwise the state of data link compression is selected based on the

determined type of data of the given protocol data unit; and

associating the selected state of data link compression with the protocol data unit to control a compression process adapted to compress contents of protocol data units.

29.-31. Cancelled.

32. The method of claim 1 wherein:

the filtering protocol-specific header and control information of a protocol data unit (PDU) to determine compressibility of the contents of said protocol data unit includes determining the type of data of the given protocol data unit where the given protocol data unit is not associated with a previously filtered protocol data unit; and

the selecting the state of data link compression for said protocol data unit to optimize compression efficiency is performed such that the state of data link compression is selected based on the determined type of data of the given protocol data unit if the given protocol data unit is not associated with a previously filtered protocol data unit.

IX. Evidence Appendix

None.

X. Related Proceedings Appendix

None.